I2RS WG D. Migault
Internet-Draft J. Halpern
Intended status: Informational Ericsson
Expires: September 8, 2017 S. Hares
Huawei
March 7, 2017

I2RS Environment Security Requirements
draft-ietf-i2rs-security-environment-reqs-03

Abstract

This document provides environment security requirements for the I2RS architecture. Environment security requirements are independent of the protocol used for I2RS. The security environment requirements are the good security practices to be used during implementation and deployment of the code related to the new interface to routing system (I2RS) so that I2RS implementations can be securely deployed and operated.

Environmental security requirements do not specify the I2RS protocol seecurity requirements. This is done in another document (draft-ietf-i2rs-protocol-security-requirements).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document provides environment security requirements for the I2RS architecture. Environment security requirements are independent of the protocol used for I2RS. The I2RS protocol security requirements [I-D.ietf-i2rs-protocol-security-requirements] define the security for the communication between I2RS client and agent. The security environment requirements are good security practices to be used during implementation and deployment of the I2RS protocol so that I2RS protocol implementations can be securely deployed and operated. These environment security requirements address the security considerations described in the I2RS Architecture [RFC7921] required to provide a stable and secure environment in which the dynamic programmatic interface to the routing system (I2RS) should operates.

Even though the I2RS protocol is mostly concerned with the interface between the I2RS client and the I2RS agent, the environmental security requirements must consider the entire I2RS architecture and specify where security functions may be hosted and what criteria should be met in order to address any new attack vectors exposed by deploying this architecture. Environment security for I2RS has to be considered the complete I2RS architecture and not only on the protocol interface.

This document is structured as follows:

- Section 2 describes the terminology used in this document,

- Section 3 describes how the I2RS plane can be securely isolated from the management plane, control plane and forwarding plane.

The subsequent sections of the document focuses on the security within the I2RS plane.

- Section 4 analyzes how the I2RS access control policies can be deployed throughout the I2RS plane in order to limit access to the routing system resources to authorized components with the authorized privileges. This analysis examines how providing a robust communication system between the components aids the access control.

- Section 5 details how I2RS keeps applications isolated from another and without affecting the I2RS components. Applications may be independent, with different scopes, owned by different tenants. In addition, the applications may modify the routing system in an automatic way.

Motivations are described before the requirements are given.

The reader is expected to be familiar with the I2RS problem statement [RFC7920], I2RS architecture, [RFC7921], traceability requirements [RFC7922], I2RS Pub/Sub requirements [RFC7923], I2RS ephemeral state requirements [I-D.ietf-i2rs-ephemeral-state], I2RS protocol security requirements [I-D.ietf-i2rs-protocol-security-requirements].

2. Terminology and Acronyms

- Environment Security Requirements : Security requirements specifying how the environment a protocol operates in needs to be secure. These requirements do not specify the protocol security requirements.

- I2RS plane: The environment the I2RS process is running on. It includes the applications, the I2RS client and the I2RS agent.

- I2RS user: The user of the I2RS client software or system.

- I2RS access control policies: The policies controlling access of the routing resources by applications. These policies are divided into policies applied by the I2RS client regarding applications and policies applied by the I2RS agent regarding I2RS clients.

- I2RS client access control policies: The access control policies processed by the I2RS client.

- I2RS agent access control policies: The access control policies processed by the I2RS agent.

2.1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. I2RS Plane Isolation

Isolating the I2RS plane from other network planes (the management, forwarding plane, and control planes) is fundamental to the security of the I2RS environment. Clearly differentiating the I2RS components from the rest of the network device does the following:

1. protects the I2RS components from vulnerabilities in other parts of the network,

2. protects other systems vital to the health of the network from vulnerabilities in the I2RS plane.

Separating the I2RS plane from other network control and forwarding planes is similar to the best common practice of placing software into software containers within modules with clear interfaces to exterior modules. In a similar way, although the I2RS plane cannot be completely isolated from other planes, it can be carefully designed so the interactions between the I2RS plane and other planes can be identified and controlled. The following is a brief description of how the I2RS plane positions itself in regard to the other planes.

3.1. I2RS Plane and Management plane

The purpose of the I2RS plane is to provide a standard programmatic interface of the routing system resources to network oriented applications. Routing protocols often run in a control plane and provide entries for the forwarding plane as shown in figure 1. The I2RS plane which contains the I2RS applications, the I2RS client, the north bound interface between the I2RS client and I2RS applications, the I2RS protocol, the I2RS agent, and south bound API (SB API) to the routing system. The communication interfaces in the I2RS plane are shown on the the left hand side of the figure 1.

The management plane contains the mangement application the management client, the north bound API between the management client and management application, the mangement server, the management protocol (E.g. RESTCONF) between mangement client and management server, and the south bound API between the management server and the control plane. The communication interfaces associated with the management plane are shown on the right hand side of figure 2.

The I2RS plane and the management plane both interact with control plane on which the routing systems operate. [RFC7921] describes several of these interaction points such as the local configuration, the static system state, routing, and signaling. A routing resource may be accessed by I2RS plane, the mangement plane, or routing protocol(s) which creates the potential for overlapping access. The southbound APIs can limit the scope of the management plane’s and the I2RS plane’s interaction with the routing resources.

Security focus:

Implementers need to carefully examine the southbound APIs from the I2RS Plane and the management plane to determine if these APIs overlap or conflict during use. If these APIs overlap or conflict, these interactions can provide errors which are not traceable or provide a risk for security intrusions between the two planes.

APIs that interact with the
I2RS Plane and Management Plane

```
     I2RS applications     Mangement applications
      || NB API                  NB API ||
      ||                                ||
      I2RS plane           management plane
      || control plane     configuration||
      || datastore            datastore ||
      ||                                ||
      ||SB API                   SB API ||
      ---------------------------------------
      | Routing System with protocols       |<protocols>
      | control plane                       |
      | (applied datastore)                 |
      +-------------------------------------+
      |  forwarding plane                   |
      +-------------------------------------+
      |   system                            |
      +-------------------------------------+
```

Figure 1 - North Bound (NB) APIs and
South Bound (SB) APIs

The north bound interface may also be a source of conflicting interactions between the I2RS plane and the management plane. It is important that conflicting interactions do not provide a deadlock situation or allow a vulnerability due to data store leaking.

3.1.1. Deadlocks

How can a deadlock occur?

The I2RS applications and the management applications may both interact with the Routing System. For example, I2RS applications may set ephemeral state for an BGP routing process allowing a peer to temporarily increase the maximum number of prefix it will accept. At the same time, a management plane process may change a BGP Peer's configuration for the maximum number of prefixes to decrease the maximum number of prefixes. [RFC7921] suggests that implementations SHOULD provide operator configurable knobs that will determine which functions (I2RS or configuration management) has precedence in the routing system, and that events should signal an I2RS agent if the I2RS ephemeral state is overwritten. This is an example of policy that prevents a deadlock situation for both the I2RS application and the mangement application.

It is important that implementations include both policy knobs for resolving the deadlocks between the the I2RS plane and the management plane, and event signaling which reports deadlocks occuring within a system supporting I2RS.

3.1.2. Data Store leaking

A vulnerability can occur if there is leakage between the I2RS ephemeral control plane data store and the management plane’s configuration datastore. [I-D.ietf-netmod-revised-datastores] describes a datastore architecture with control plane datastores (such as the I2RS protocol’s ephemeral datastore) being logically different than the the configuration data store. The mixture of the I2RS ephemeral configuration and management configuration is done by the routing system code (specific to an implementation). The routing system code resolves any conflict between I2RS control plane configuration and the management plane configuration, and then installs the state in the routing system. Implementation policy determines which configuration state should be installed.

The applied datastore provides information on what is installed in each part of a system - including the routing system. The operational state data store provides both the applied data store information and additional operational state from the control plane protocols and control plane datastore.

Since it is the routing system code which mixing the configuration from the I2RS control plane datastore and the management datastore to create applied datastore for the routing system, this code must take care to prevent:

- o the I2RS system "infecting" the management system configuration datastore with information from the I2RS control plane data store,
- o the management system "infecting" the I2RS system data with data not specifically imported by I2RS data models,
- o the management system indirectly "infecting" the I2RS system by propagating improper information from one system to another via I2RS.

In this circumstance, we define "infecting" as inteferring with and leading into a incoherent state. Planned interactions such as interactions denoted in in two cooperating Yang data modules is not incoherent state.

For example, BGP configuration and BGP I2RS ephemeral state configuration could have a defined interaction. The I2RS plane may legitimately update a routing resource configured by the management plane, or the reverse (the management plane updates a resource the I2RS plane has configured) if the interactions are defined by Yang modules or local policy. Infecting, is when the implementation is updated by two processes resulting in an incoherent state. Indirect "infecting", we define as as changes made by the I2RS plane that cause changes in routing protocols which add state unexpected by the management plane or the reverse (the mangement plane adding changes in routing protocols the I2RS plane does not expect).

Prevention for Data Store Leakage

The primary protection in this space is going to need to be validation rules on:

- the data being sent/received by the I2RS agent (including notification of changes that the I2RS agent sends the I2RS client),
- any data transferred between management datastores (configuration or operational state) and I2RS ephemeral control plane data stores;
- data transferred between I2RS Agent and Routing system,
- data transferred between a management server and the I2RS routing system,
- data transfered between I2RS agent and system (e.g. interfaces ephemeral configuration),
- data transferred between management server and the system (e.g. interface configuration).

The next few paragraphs will provide some ideas on how this might be implemented. These suggest implementation policy should resolve what is not resolved in the YANG Data module definitions.

The Network Access Control Module (NACM) has been define to control access to the configuration datastore via the management protocol across a network. A similar network access control module could be defined for the I2RS-NACM (per [I-D.ietf-netconf-rfc6536bis]. Figure 2 shows how the I2RS-NACM could be created to support parallel features with the management protocol (E.g. NETCONF) NACM.

I2RS implementations may also need to define an access modules which control access to the routing system (Routing Access Control Module (RACM)) by policy. The I2RS-RACM would control how the I2RS agent access the routing system through the SB API interface. In parallel, the management system would have a RACM to control the SB API interface (see figure 2). I2RS agent and the management server may want to read/write system information interfaces or other system functions. In order to prevent leakage between the I2RS system and the management system, there needs to be system access control module (SACM) that protects the jointly held data.

```
   I2RS- || (I2RS          Mgt protocol  ||
   NACM  ||  Protocol)     (E.g. NETCONF)|| NACM
   -------                         ----------
    I2RS ||I2RS Agent          Mgt server||
    SACM || I2RS Control-DS    config DS || SACM
    -----|| (RACM)                (RACM) ||=======||
    |    ||SB API                 SB API ||       ||
    |  +-----------------------------------+      ||
    |  | Routing System with protocols     |      ||
    |  | control plane                     |      ||
    |  | (applied datastore)               |      ||
    |  | (operational state)               |      ||
    |  +--------------||-------------------+      ||
    |  |  forwarding plane                 |      ||
    |  +--------------||-------------------+      ||
    ---|   system                          |======||
       +-----------------------------------+


    *Mgt = management
     DS = Datastore
     Control-DS = Control plane protocol data store
     NACM = Network Access Control Module
     RACM = Routing System Access Control Module
```

figure 2

The I2RS clients and the I2RS agents also need a set of policy which defines what can be received from the network or sent to the network.

```
     I2RS applications       Mangement applications
      || NB API                    MB API ||
      || (APP NAC policy) (APP NAC policy)||
      || I2RS client          mgt client ||
      || (NAC policy)        (NAC policy)||
      ||                                  ||
    I2RS protocol               mgt protocol
                            (NETCONF/RESTCONF)
```

figure 3

3.2. I2RS Plane and Forwarding Plane

Applications hosted by the I2RS client belong to the I2RS plane. It is difficult to constrained these applications to the I2RS plane, or even to limit their scope within the I2RS plane. Applications using I2RS may also interact with components outside the I2RS plane. For example an application may use an management client to configure the network and monitored events via an I2RS agent as figure 4 shows.

```
    +--------------------------------------+
    |       Application                     |
    +--------------------------------------+
      || NB API                NB API ||
      || I2RS client       mgt client ||
      ||                              ||
       I2RS protocol       mgt protocol
                            (NETCONF/RESTCONF)
```

figure 4

Applications may also communicate with multiple I2RS clients in order to have a broader view of the current and potential states of the network and the I2RS plane itself. These varied remote communication relationships between applications using the I2RS protocol to change the forwarding plane make it possible for an individual application to be an effective attack vector against the operation of the network, a router’s I2RS plane, the forwarding plane of the routing system, and other planes (management and control planes).

Prevention measures:

Systems should consider the following prevention errors:

application validation - There is little the I2RS plane can do to validate applications with which it interacts. The I2RS client passes the I2RS agent an opaque identifier for the application so that an application’s actions can be traced back to the application.

Validation against common misconfigurations or errors - One way of securing the interfaces between application, the I2RS plane, and the forwarding plane is to limit the information accepted and to limit the rate information is accepted between these three software planes. Another method is to performing rudimentary checks on the results of any updates to the forwarding plane.

3.3. I2RS Plane and Control Plane

The network control plane consists of the processes and protocols that discover topology, advertise reachability, and determine the shortest path between any location on the network and any destination. I2RS client configures, monitors or receives events via the I2RS agent’s interaction with the routing system including the process that handling the control plane signaling protocols (BGP, ISIS, OSPF, etc.), route information databases (RIBs), and interface databases. In some situation to manage an network outage or to control traffic, the I2RS protocol may modify information in the route database or the configuration of routing process. While this is not a part of normal processing, such action allows the network operator to bypass temporary outages or DoS attacks.

This capability to modify the routing process information carries with it the risk that the I2RS agent may alter the normal properties of the routing protocols which provide normal loop free routing in the control plane. For example, information configured by the I2RS agent into routing process or RIBs could cause forwarding problems or routing loops. As a second example, state which is inserted or deleted from routing processes (control traffic, counters, etc.) could cause the routing protocols to fail to converge or loop).

Prevention measures:

The I2RS implementation can provide internal checks that after a routing system protocol changes that it is still operating correctly. These checks would be specific to the routing protocol the I2RS Agent would change. For example, if a BGP maximum prefix limit for a BGP peer is lowered then the BGP peer should not allow the number prefixes received from that peer to exceed this number.

3.4. Requirements

To isolate I2RS transactions from other planes, it is required that:

SEC-ENV-REQ 1: Application-to-routing system resources communications should use an isolated communication channel. Various level of isolation can be considered. The highest level of isolation may be provided by using a physically isolated network. Alternatives may also consider logical isolation (e.g. using vLAN). In a virtual environment that shares a common infrastructure, encryption may also be used as a way to enforce isolation. Encryption can be added by using a secure transport required by the I2RS protocol security [I-D.ietf-i2rs-protocol-security-requirements], and sending the non-confidential I2RS data (designed for a non-secure transport) over a secure transport.

SEC-ENV-REQ 2: The interface used by the routing element to receive I2RS transactions via the I2RS protocol (e.g. the IP address) SHOULD be a dedicated physical or logical interface. As previously, mentioned a dedicated physical interface may contribute to a higher isolation. Isolation may also be achieved by using a dedicated IP address or a dedicated port.

SEC-ENV-REQ 3: An I2RS agent SHOULD have specific permissions for interaction with each routing element and access to the routing element should governed by policy specific to the I2RS agent’s interfaces (network, routing system, system, or cross-datastore).

Explanation:

When the I2RS agent performs an action on a routing element, the action is performed in a process (or processes) associated with a routing process. For example, in a typical UNIX system, the user is designated with a user id (uid) and belongs to groups designated by group ids (gid). If such a user id (uid) and group id (gid) is the identifier for the routing processes peforming routing tasks in the control plane, then the I2RS Agent process would communicate with these routing processes. It is important that the I2RS agent has its own unique identifier and the routing processes have their own identifier so that access control can uniquely filter data between I2RS Agent and routing processes.

The specific policy the the I2RS agent uses to filter data from the network or from different processes on a system (routing, system or cross-datastore) should be specific to the I2RS agent. For example, the network access filter policy that the I2RS agent uses should be uniquely identifiable from the configuration datastore updated by a management protocol.

SEC-ENV-REQ 4: I2RS plane should be informed when a routing system resource is modified by a user outside the I2RS plane access. Notifications from the control plane SHOULD not to flood the I2RS plane, and rate limiting (or summarization) is expected to be applied. These routing system notification MAY translated to the appropriate I2RS agent notifications, and passed to various I2RS clients via notification relays.

Explanation:

This requirements is also described in section 7.6 of [RFC7921] for the I2RS client, and this section extends it to the entire I2RS plane (I2RS agent, client, and application).

A routing system resource may be accessed by the management plane or control plane protocols so a change to a routing system resource may remain unnoticed unless and until the routing system resource notifies the I2RS plane by notifying the I2RS agent. Such notification is expected to trigger synchronization of the I2RS resource state between the I2RS agent and I2RS client - signalled by the I2RS agent sending a notification to an I2RS client.

The updated resource should be available in the operational state if there is a yang module referencing that operational state, but this is not always the case. In the cases, where operational state is not updated, the I2RS SB (southbound) API should include the ability to send a notification.

SEC-ENV-REQ 5: I2RS plane should define an "I2RS plane overwrite policy". Such policy defines how an I2RS is able to update and overwrite a resource set by a user outside the I2RS plane. Such hierarchy has been described in section 6.3 and 7.8 of [RFC7921]

Explanation:

A key part of the I2RS architecture is notification regarding routing system changes across the I2RS plane (I2RS client to/from I2RS agent). The security environment requirements above (SEC-ENV-REQ-03 to SEC-ENV-REQ-05) provide the assurance that the I2RS plane and the routing systems the I2RS plane attaches to remains untouched by the other planes or the I2RS plane is notificed of such changes. Section 6.3 of [RFC7921] describes how the I2RS agent within the I2RS plane interacts with forwarding plane’s local configuration, and provides the example of an overwrite policy between the I2RS plane and local configuration (instantiated in 2 Policy Knobs) that operators may wish to set. The prompt notification of any outside overwrite is key to the architecture and proper interworking of the I2RS Plane.

4. I2RS Access Control for Routing System Resources

This section provides recommendations on how I2RS access control policies associated to the routing system resources. These policies only apply within the I2RS plane. More especially, the policies are associated to the applications, I2RS clients and I2RS agents, with their associated identity and roles.

The I2RS deployment of I2RS applications, I2RS clients, and I2RS agents can be located locally in a closed environment or distributed over open networks. The normal case for routing system management is over an open environment. Even in a closed environment access control policies should be carefully defined to be able to, in the future to carefully extend the I2RS plane to remote applications or remote I2RS clients.

[I-D.ietf-i2rs-protocol-security-requirements] defines the security requirements of the I2RS protocol between the I2RS client and the I2RS agent over a secure transport. This section focuses on I2RS access control architecture (section 4.1), access control policies of the I2RS agent (section 4.2), the I2RS client (section 4.3), and the application (section 4.4).

4.1. I2RS Access Control Architecture

Overview:

Applications access to routing system resource via numerous intermediaries nodes. The application communicates with an I2RS client. In some cases, the I2RS client is only associated to a single application attached to one or more agents (case a and case b in figure 4 below). In other cases, the I2RS client may be connected to two applications (case c in figure 4 below), or the I2RS may act as a broker (agent/client device shown in case d in the figure 4 below). The I2RS client broker approach provides scalability to the I2RS architecture as it avoids that each application be registered to the I2RS agent. Similarly, the I2RS access control should be able to scale numerous applications.

The goal of the security environment requirements in this section are to control the interactions between the applications and the I2RS client, and the interactions between the I2RS client and the I2RS agent. The key challenge is that the I2RS architecture puts the I2RS Client in control of the stream of communication (application to I2RS client and I2RS client to the I2RS agent). The I2RS agent must trust the I2RS client’s actions without having an ability to verify the I2RS client’s actions.

a) I2RS application-client pair talking
   to one I2RS agent

```
+-----------+     +---------+      +-------+
|   I2RS    |=====| I2RS    |======| I2RS  |
|application|     | client 1|      | agent |
+-----------+     +---------+      +-------+
```

b) I2RS application client pair talking to
   two i2RS agents

```
                                  +--------+
+-------------+   +---------+     | I2RS   |
|   I2RS      |===| I2RS    |=====| agent 1|
|application 1|   | client 1|     +--------+
|             |   |         |     +--------+
|             |   |         |=====| I2RS   |
+-------------+   +---------+     | agent 2|
                                  +--------+
```

c) two applications talk to 1 client

```
                                 +--------+
+-------------+   +--------+     | I2RS   |
|   I2RS      |===|I2RS    |=====| agent 1|
|application 1|   |client 1|     +--------+
+-------------+   |        |     +--------+
+-------------+   |        |=====| I2RS   |
|  I2RS       |   |        |     | agent 2|
|application 2|===|        |     +--------+
+-------------+   +--------+
```

d) I2RS Broker (agent/client

```
                                +--------+
+-------------+  +--------+     | I2RS   |
|   I2RS      |==|I2RS    |=====|agent 3/|
|application 1|  |client 1|   ==|client 3+----+
+-------------+  +--------+   | +--+-----+    |
                              |    |         |
+-------------+  +--------+   |  +-------+ +--|----+
|  I2RS       |  |I2RS    |===|  |I2RS   | |I2RS   |
|application 2|==|client 2|      |agent 1| |agent 2|
+-------------+  +--------+      +-------+ +-------+
```

figure 4

4.1.1. access control Enforcement Scope

SEC-ENV-REQ 6: I2RS access control should be performed through the whole I2RS plane. It should not be enforced by the I2RS agent only within the routing element. Instead, the I2RS client should enforce the I2RS client access control against applications and the I2RS agent should enforce the I2RS agent access control against the I2RS clients. The mechanisms for the I2RS client access control are not in scope of the I2RS architecture [RFC7921], which exclusively focuses on the I2RS agent access control provided by the I2RS protocol.

Explanation:

This architecture results in a layered and hierarchical or multi-party I2RS access control. An application will be able to access a routing system resource only if both the I2RS client is granted access by the I2RS agent and the application is granted access by the I2RS client.

4.1.2. Notification Requirements

SEC-ENV-REQ 7: When an access request to a routing resource is refused by one party (the I2RS client or the I2RS agent), the requester (e.g the application) as well as all intermediaries should indicate the reason the access has not been granted, and which entity rejected the request.

Explanation:

In case the I2RS client access control or the I2RS agent access control does not grant access to a routing system resource, the application should be able to determine whether its request has been rejected by the I2RS client or the I2RS agent as well as the reason that caused the reject.

SEC-REQ-07 indicates the I2RS agent may reject the request because the I2RS client is not an authorized I2RS client or lacks the privileges to perform the requested transaction (read, write, start notifications or logging). The I2RS client should be notified of the reason the I2RS agent rejected the transaction due to a lack of authorization or priviledges, and the I2RS client should return a message to the application indicating the I2RS agent reject the transacation with an indication of this reason. Similarly, if the I2RS client does not grant the access to the application, the I2RS client should also inform the application. An example of an error message could be, "Read failure: you do not have the read permission", "Write failure: you do not have write permission", or "Write failure: resource accessed by someone else".

This requirement has been written in a generic manner as it concerns the following interactions:

- interactions between the application and the I2RS client,

- interactions between the I2RS client and the I2RS agent at a content level (Protocol security requirements are described by [I-D.ietf-i2rs-protocol-security-requirements]), and

- interactions between the I2RS agent and the I2RS routing system (forwarding plane, control plane, and routing plane).

4.1.3. Trust

SEC-ENV-REQ 8: In order to provide coherent access control policies enforced by multiple parties (e.g. the I2RS client or the I2RS agent), theses parties should trust each others, and communication between them should also be trusted (e.g. TLS) in order to reduce additional vector of attacks.

SEC-ENV-REQ 9: I2RS client or I2RS agent SHOULD also be able to refuse a communication with an application or an I2RS client when the communication channel does not fulfill enough security requirements.

Explanation:

The participants in the I2RS Plane (I2RS client, I2RS agent, and I2RS application) exchange critical information, and to be effective the communication should be trusted and free from security attacks.

The I2RS client or the I2RS agent should be able to reject messages over a communication channel that can be easily hijacked, like a clear text UDP channel.

4.1.4. Sharing access control Information

For the I2RS client:

SEC-ENV-REQ 10: The I2RS client MAY request information on its I2RS access control subset policies from the I2RS agent or cache requests that have been rejected by the I2RS agent to limit forwarding unnecessary queries to the I2RS agent.

SEC-ENV-REQ 11: The I2RS client MAY support receiving notifications when its I2RS access control subset policies have been updated by the I2RS agent.

Similarly, for the applications:

SEC-ENV-REQ 12: The applications MAY request information on its I2RS access control subset policies in order to limit forwarding unnecessary queries to the I2RS client.

SEC-ENV-REQ 13: The applications MAY subscribe to a service that provides notification when its I2RS access control subset policies have been updated.

For both the application and the client:

SEC-ENV-REQ 14: The I2RS access control should explicitly specify accesses that are granted. More specifically, anything not explicitly granted should be denied (default rule).

Explanation:

In order to limit the number of access requests that result in an error, each application or I2RS client can retrieve the I2RS access control policies that applies to it. This subset of rules is designated as the "Individual I2RS access control policies". As these policies are subject to changes, a dynamic synchronization mechanism should be provided. However, such mechanism may be implemented with different level of completeness and dynamicity of the individual I2RS access control policies. One example, may be caching transaction requests that have been rejected.

I2RS access control should be appropriately be balanced between the I2RS client and the I2RS agent. It remains relatively easy to avoid the complete disclosure of the access control policies of the I2RS agent. Relative disclosure of access control policies may allow the leaking confidential information in case of misconfiguration. It is important to balance the level of trust of the I2RS client and the necessity of distributing the enforcement of the access control policies.

I2RS access control should not solely rely only on the I2RS client or the I2RS agent as illustrated below:

- 1) I2RS clients are dedicated to a single application: In this case, it is likely that I2RS access control is enforced only by the I2RS agent, as the I2RS client is likely to accept all access request of the application. It is recommended that even in this case, I2RS client access control is not based on an "Allow anything from application" policy, but instead the I2RS client specifies accesses that are enabled. In addition, the I2RS client may sync its associated I2RS access control policies with the I2RS agent to limit the number of refused access requests being sent to the I2RS agent. The I2RS client is expected to balance benefits and problems with synchronizing its access control policies with the I2RS agent to proxy request validation versus simply passing the access request to the I2RS agent.

- 2) A single I2RS client connects to multiple applications or acts as a broker for many applications:
  In this case the I2RS agent has a single I2RS client attached, so the I2RS client could end being configured to enforce access control policies instead of the I2RS Agent. In this circumstance, it is possible that the I2RS agent may grant an I2RS client with high priviledges and blindly trust the I2RS client without enforcing access control policies on what the I2RS client can do. Such a situation must be avoided as it could be used by malicious applications for a priviledge escalation by compromising the I2RS client causing the I2RS client to perform some action on behalf of the application that it normally does not have the priviledges to perform. In order to mitigate such attack, the I2RS client that connects to multiple applications or operates as a broker is expected to host application with an equivalent level of privileges.

4.1.5. Sharing Access Control in Groups of I2RS Clients and Agents

Overview:

To distribute the I2RS access control policies between I2RS clients and I2RS agents, I2RS access control policies can also be distributed within a set of I2RS clients or a set of I2RS agents.

Requirements:

SEC-ENV-REQ 15: I2RS clients should be distributed and act as brokers for applications that share roughly similar permissions.

SEC-ENV-REQ 16: I2RS agents should be avoided granting extra privileges to their authorized I2RS client. I2RS agent should be shared by I2RS client with roughly similar permissions. More explicitly, an I2RS agent shared between I2RS clients that are only provided read access to the routing system resources does not need to perform any write access, so the I2RS client should not be provided these accesses.

SEC-ENV-REQ 17: I2RS client and I2RS agent should be able to trace [RFC7922] the various transaction they perform as well as suspicious activities. These logs should be collected regularly and analyzed by functions that may be out of the I2RS plane.

Explanation:

This restriction for distributed I2RS clients to act as brokers only for applications with roughly the same priviledges avoids the I2RS client extra priviledges compared to hosted applications, and discourages applications from perform privilege escalation within an I2RS client. For example, suppose an I2RS client requires write access to the resources. It is not recommended to grant the I2RS agent the write access in order to satisfy a unique I2RS client. Instead, the I2RS client that requires write access should be connected to a I2RS agent that is already shared by I2RS client that requires a write access.

Access control policies enforcement should be monitored in order to detect violation of the policies or detect an attack. Access control policies enforcement may not be performed by the I2RS client or the I2RS agent as violation may require a more global view of the I2RS access control policies. As a result, consistency check and mitigation may instead be performed by the management plane. However, I2RS clients and I2RS agents play a central role.

The I2RS agent can trace transactions that an I2RS client requests it to perform, and to link this to the application via the secondary opaque identifier to the application. This information is placed in a tracing log which is retrieved by management processes. If a particular application is granted a level of priviledges it should not have, then this tracing mechanism may detect this security intrusion after the instrusion has occurred.

4.1.6. Managing Access Control Policy

Access control policies should be implemented so that the policies remain manageable in short and longer term deployments of the I2RS protocol and the I2RS plane.

Requirements:

SEC-ENV-REQ 18: access control should be managed in an automated way, that is granting or revoking an application should not involve manual configuration over the I2RS plane (I2RS client, I2RS agent, and application).

Explanation:

Granting or configuring an application with new policy should not require manual configuration of I2RS clients, I2RS agents, or other applications.

SEC-ENV-REQ 19: Access control should be scalable when the number of application grows as well as when the number of I2RS client increases.

Explanation:

A typical implementation of a local I2RS client access control policies may result in creating manually a system user associated to each application. Such an approach is likely not to scale when the number of applications increases into the hundreds.

SEC-ENV-REQ 20: Access control should be dynamically managed and easily updated.

Explanation:

Although the numberof I2RS clients is expected to be lower than the number of application, as I2RS agent provide access to the routing resource, it is of primary importance that an access can be granted or revoke in an efficient way.

SEC-ENV-REQ 21: I2RS clients and I2RS agents should be uniquely identified in the network to enable centralized management of the I2RS access control policies.

Explanation:

Centralized management of the access control policies of an I2RS plane with network that hosts several I2RS applications, clients and agents requires that each devices can be identified.

4.2. I2RS Agent Access Control Policies

Overview:

The I2RS agent access control restricts the routing system resource access to authorized identities - possible access policies may be none, read or write. The initiator of an access request to a routing resource is always an application. However, it remains challenging for the I2RS agent to establish its access control policies based on the application that initiates the request.

First, when an I2RS client acts as a broker, the I2RS agent may not be able to authenticate the application. In that sense, the I2RS agent relies on the capability of the I2RS client to authenticate the applications and apply the appropriated I2RS client access control.

Second, an I2RS agent may not uniquely identify a piece of software implementing an I2RS client. In fact, an I2RS client may be provided multiple identities which can be associated to different roles or privileges. The I2RS client is left responsible for using them appropriately according to the application.

Third, each I2RS client may contact various I2RS agent with different privileges and access control policies.

4.2.1. I2RS Agent Access Control

This section provides recommendations on the I2RS agent access control policies to keep I2RS access control coherent within the I2RS plane.

Requirements:

SEC-ENV-REQ 22: I2RS agent access control policies should be primarily based on the I2RS clients as described in [RFC7921].

SEC-ENV-REQ 23: I2RS agent access control policies MAY be based on the application if the application identity has been authenticated by the I2RS client and passed via the secondary identity to the I2RS agent.

SEC-ENV-REQ 24: The I2RS agent should know which identity (E.g. system user) performed the latest update of the routing resource. This is true for an identity inside and outside the I2RS plane so the I2RS agent can appropriately perform an update according to the priorities associated to the requesting identity and the identity that last updated the resource.

SEC-ENV-REQ 25: the I2RS agent should have a "I2RS agent overwrite Policy" that indicates how identities can be prioritized. This requirements is also described in section 7.6 of [RFC7921]. Similar requirements exist for components within the I2RS plane, but this is within the scope of the I2RS protocol security requirements [I-D.ietf-i2rs-protocol-security-requirements].

Explanation:

If the I2RS application is authenticated to the I2RS client, and the I2RS client is authenticated to the I2RS agent, and the I2RS client uses the opaque secondary identifier to pass an authenticated identifier to the I2RS agent, then this identifier may be used for access control. However, caution should be taken when using this chain of authentication since the secondary identifier is intended in the I2RS protocol only to aid traceability.

From the environment perspective the I2RS agent MUST be aware when the resource has been modified outside the I2RS plane by another plane (management, control, or forwarding). The prioritization between the different planes should set a deterministic policy that allows the collision of two planes (I2RS plane and another plane) to be resolved via an overwrite policy in the I2RS agent.

Similar requirements exist for knowledge about identities within the I2RS plane which modify things in the routing system, but this is within the scope of the I2RS protocol’s requirements for ephemeral state [I-D.ietf-i2rs-ephemeral-state] and security requirements [I-D.ietf-i2rs-protocol-security-requirements].

4.2.2. I2RS Client Access Control Policies

Overview:

The I2RS client access control policies are responsible for authenticating the application managing the privileges for the applications, and enforcing access control to resources by the applications.

Requirements:

REQ 26: I2RS client should authenticate its applications. If the I2RS client acts as a broker and supports multiple applications, it should authenticate each application.

REQ 27: I2RS client should define access control policies associated to each applications. An access to a routing resource by an application should not be forwarded immediately and transparently by the I2RS client based on the I2RS agent access control policies. The I2RS client should first check whether the application has sufficient privileges, and if so send an access request to the I2RS agent.

Explanation:

If no authentication mechanisms have being provided between the I2RS client and the application, then I2RS client must be dedicated to a single application. By doing so, application authentication relies on the I2RS authentication mechanisms between the I2RS client and the I2RS agent.

If an I2RS client has multiple identities that are associated with different privileges for accessing an I2RS agent(s), the I2RS client access control policies should specify the I2RS client identity with the access control policy.

4.2.3. Application and Access Control Policies

Overview

Applications do not enforce access control policies. Instead these are enforced by the I2RS clients and the I2RS agents. This section provides recommendations for applications in order to ease I2RS access control by the I2RS client and the I2RS agent.

Requirements:

SEC-ENV-REQ 28: Applications SHOULD be uniquely identified by their associated I2RS clients

Explanation:

Different application may use different methods (or multiple methods) to communicate with its associated I2RS client, and each application may not use the same form of an application identifier. However, the I2RS client must obtain an identifier for each application. One method for this identification can be a system user id.

SEC-ENV-REQ 29: Each application SHOULD be associated to a restricted number of I2RS client

Explanation:

The I2RS client provides access to resource on its behalf and this access should only be granted for trusted applications, or applications with an similar level of trust. This does not prevent an I2RS client to host a large number of applications with the same levels of trust.

SEC-ENV-REQ 30: An application SHOULD be provided means and methods to contact their associated I2RS client.

Explanation:

It is obvious when an I2RS client belongs to the application as part of a module or a library that the application can communicate with a I2RS client. Similarly, if the application runs into a dedicated system with a I2RS client, it is obvious which I2RS client the application should contact. If the application connects to the I2RS client remotely, the application needs some means to retrieve the necessary information to contact its associated I2RS client (e.g. an IP address or a FQDN).

5. I2RS Application Isolation

A key aspect of the I2RS architecture is the network oriented application that uses the I2RS high bandwidth programmatic interface to monitor or change one or more routing systems. I2RS applications could be control by a single entity or serve various tenants of the network. If multiple entities use an I2RS application to monitor or change the network, security policies must preserve the isolation of each entity’s control and not let malicious entities controlling one I2RS application interfere with other I2RS applications.

This section discusses both security aspects related to programmability as well as application isolation in the I2RS architecture.

5.1. Robustness Toward Programmability

Overview

I2RS provides a programmatic interface in and out of the Internet routing system which provides the following advantages for security:

- the use of automation reduces configuration errors;

- the programmatic interface enables fast network reconfiguration and agility in adapting to network attacks; and

- monitoring facilities to detect a network attack, and configuration changes which can help mitigate the network attack.

Programmability allows applications to flexible control which may cause problems due to:

- applications which belong to different tenants with different objectives,

- applications which lack coordination resulting in unstable routing configurations such as oscillations between network configurations, and creation of loops for example. For example, one application may monitor a state and change to positive, and a second application performs the reverse operation (turns it negative). This fluctuation can cause a routing system to become unstable.

The I2RS plane requires data and application isolation to prevent such situations to happen. However, to guarantee the network stability constant monitoring and error detection are recommended to be activated.

Requirement:

SEC-ENV-REQ 31: The I2RS agents should monitor constantly parts of the system for which I2RS clients or applications have provided requests. It should also be able to detect any I2RS clients or applications causing problems that may lead the routing system in an unstable state.

Explanation:

Monitoring consists at least in logging events and receiving streams of data. I2RS Plane implementations should monitor the I2RS applications and I2RS clients for potential problems. The cause for the I2RS clients or applications providing problematic requests can be failures in the implementation code or malicious intent.]

5.2. Application Isolation

5.2.1. DoS

Overview:

Requirements for robustness to DoS attacks have been addressed in the communication channel section [RFC7921]. This section focuses on requirements for application isolation that help prevent DoS.

Requirements:

SEC-ENV-REQ 32: In order to prevent DoS, it is recommended the I2RS agent controls the resources allocated to each I2RS clients. I2RS client that acts as broker may not be protected as efficiently against these attacks unless the broker performs resource controls for the hosted applications.

SEC-ENV-REQ 33: I2RS agent SHOULD make a response redirection unless the redirection is previously validated and agreed by the destination.

SEC-ENV-REQ 34: I2RS Appications should avoid the use of underlying protocols that are not robust to reflection attacks.

Explanation:

The I2RS interface is used by application to interact with the routing states. If the I2RS client is shared between multiple applications, one application can use the I2RS client to perform DoS or DDoS attacks on the I2RS agent(s) and through the I2RS agents attacks on the network. DoS attack targeting the I2RS agent would consist in providing requests that keep the I2RS agent busy for a long time. These attacks on the I2RS agent may involve an application (requesting through an I2RS Client) heavy computation by the I2RS agent in order to block operations like disk access.

Some DoS attacks may attack the I2RS Client’s reception of notification and monitoring data stream over the network. Other DoS attacks may focus on the application directly by performing reflection attacks to reflect traffic. In such an attack could be performed by first detecting an application is related to monitoring the RIB or changing the RIB. Reflection-based DoS may be also attack at various levels in the stack utilizing UDP at the service to redirect data to a specific repository

I2RS implementation should consider how to protect I2RS against such attacks.

5.2.2. Application Logic Control

Overview

This section examines how application logic must be design to ensure application isolation.

Requirements:

SEC-ENV-REQ 35: Application logic should remain opaque to external listeners. Application logic may be partly hidden by encrypting the communication between the I2RS client and the I2RS agent. Additional ways to obfuscate the communications may involve sending random messages of various sizes. Such strategies have to be balanced with network load. Note that I2RS client broker are more likely to hide the application logic compared to I2RS client associated to a single application.

Explanation:

Applications use the I2RS interface in order to update the routing system. These updates may be driven by behavior on the forwarding plane or any external behaviors. In this case, correlating observation to the I2RS traffic may enable to derive the application logic. Once the application logic has been derived, a malicious application may generate traffic or any event in the network in order to activate the alternate application.

6. Security Considerations

The whole document is about security requirements for the I2RS environment. To protect personal privacy, any identifier (I2RS application identifier, I2RS client identifier, or I2RS agent identifier) should not contain personal identifiable information.

7. IANA Considerations

No IANA considerations for this requirements.

8. Acknowledgments

A number of people provided a significant amount of helping comments and reviews. Among them the authors would like to thank Russ White, Russ Housley, Thomas Nadeau, Juergen Schoenwaelder, Jeffrey Haas, Alia Atlas, and Linda Dunbar.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC7920] Atlas, A., Ed., Nadeau, T., Ed., and D. Ward, "Problem Statement for the Interface to the Routing System", RFC 7920, DOI 10.17487/RFC7920, June 2016, <http://www.rfc-editor.org/info/rfc7920>.

[RFC7921] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", RFC 7921, DOI 10.17487/RFC7921, June 2016, <http://www.rfc-editor.org/info/rfc7921>.

[RFC7922] Clarke, J., Salgueiro, G., and C. Pignataro, "Interface to the Routing System (I2RS) Traceability: Framework and Information Model", RFC 7922, DOI 10.17487/RFC7922, June 2016, <http://www.rfc-editor.org/info/rfc7922>.

[RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", RFC 7923, DOI 10.17487/RFC7923, June 2016, <http://www.rfc-editor.org/info/rfc7923>.

[I-D.ietf-i2rs-protocol-security-requirements]
Hares, S., Migault, D., and J. Halpern, "I2RS Security Related Requirements", draft-ietf-i2rs-protocol-security-requirements-17 (work in progress), September 2016.

9.2. Informative References

[I-D.ietf-i2rs-ephemeral-state]
Haas, J. and S. Hares, "I2RS Ephemeral State Requirements", draft-ietf-i2rs-ephemeral-state-23 (work in progress), November 2016.

[I-D.ietf-netconf-rfc6536bis]
Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", draft-ietf-netconf-rfc6536bis-00 (work in progress), January 2017.

[I-D.ietf-netmod-revised-datastores]
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "A Revised Conceptual Model for YANG Datastores", draft-ietf-netmod-revised-datastores-00 (work in progress), December 2016.

Authors' Addresses

Daniel Migault
Ericsson
8400 boulevard Decarie
Montreal, QC   H4P 2N2
Canada

Phone: +1 514-452-2160
Email: daniel.migault@ericsson.com

Joel Halpern
Ericsson

Email: Joel.Halpern@ericsson.com

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com